VASS-AKA: An Efficient Batch Verification Protocol for Value Added Services

Neetesh Saxena
Department of CSE, IIT Indore, India
neetesh.saxena@iiti.ac.in

Abstract— In this paper, we introduce an efficient batch oriented authentication and key agreement protocol for value added services named VASS-AKA to authenticate multiple requests sent from different mobile stations at a time. The reliability of the protocol is discussed along with the algorithm to detect one or more invalid mobile user(s) in a batch. The security analysis and performance evaluation in terms of communication and computation overhead, batch and re-batch verification delay is presented. Using this protocol, the authentication server efficiently able to verify authentication requests sent from multiple mobile users.

Index Terms—Authentication, batch verification, security, VAS

I. INTRODUCTION

Presently, mobile computing and telecommunications industries are one of the fastest growing industries worldwide. These technologies are useful in environments where a mobile user purchase value added services (VAS) through a value-added service provider (VASP). However, this mobile scenario places a number of restrictions on the design of suitable security protocols. Most significantly, a mobile user typically has limited computational capabilities compared to an entity in a fixed network. Thus the necessary computational effort made by any protocol should be minimized. It is expected that in the future, the constraints related to VASP’s performance may arise which may lead to a bottleneck during multiple authentication requests at a time. For this reason, we also focus on the computational effort required by the VASP, seeking to minimize it as much as possible. Moreover, the communications bandwidth between the user and VASP may be limited, hence the size of messages used for communication in the protocol should be kept as small as possible.

A. Existing Solutions

There are many solutions have been proposed by various researchers for the value added services in vehicular ad hoc networks [1], [2], [3] [4], [5] and social networks [6]. However, in the literature, we did not find any batch verification protocol, providing value added services to the mobile users. Thus, it is strongly required to implement such application oriented protocols to facilitate the mobile users with different value added services where server receives requests in bulk at a time such as m-commerce, e-ticket booking services like railway reservation, match tickets etc. through the mobile phone.

B. Our Contribution

In this paper, we propose and present an efficient authentication and key agreement protocol named VASS-AKA for proving value added services to the mobile users. The highlights of the proposed protocol are as follows:

1. This protocol handles multiple authentication requests at a time or in a fixed (very less) time duration.
2. The protocol provides mutual authentication between each mobile station (MS) and the authentication server (AS).
3. The original identity of the MS is kept private during the transmission of information over the network.
4. In one way multiple/single authentication request(s) process from device to the server, the VASS-AKA protocol reduces 11.12%, 33.34%, 61.65%, and 66.47% transmission bandwidth as compared to the IBV, ABAKA, BLS and ECDSA-AKA protocols respectively.
5. From server to the device one way multiple/single authentication request(s) process, the VASS-AKA protocol reduces the communication bandwidth as 80% and 90.5% as compared to the ABAKA and ECDSA-AKA protocols.

II. SYSTEM, COMMUNICATION, AND SECURITY MODEL

This section presents a system model and the basic preliminaries required in order to develop an efficient protocol for the value added services.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Size (Bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSI</td>
<td>International Mobile Subscriber Identity</td>
<td>128</td>
</tr>
<tr>
<td>TID</td>
<td>Temporary Identity</td>
<td>128</td>
</tr>
<tr>
<td>G</td>
<td>Identity of Service Provider</td>
<td>128</td>
</tr>
<tr>
<td>SK</td>
<td>Shared secret key between MS and the AS (AuC)</td>
<td>128</td>
</tr>
<tr>
<td>DK</td>
<td>Delegation key generated from SK</td>
<td>128</td>
</tr>
<tr>
<td>T</td>
<td>Timestamp</td>
<td>64</td>
</tr>
<tr>
<td>k</td>
<td>A random number</td>
<td>128</td>
</tr>
<tr>
<td>S</td>
<td>Signature generated by MS</td>
<td>128</td>
</tr>
<tr>
<td>f1()</td>
<td>Function generated by MS</td>
<td>-</td>
</tr>
<tr>
<td>f2()</td>
<td>Function generated by TID</td>
<td>-</td>
</tr>
<tr>
<td>(.)</td>
<td>Bitwise X-OR operation</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The remainder of this paper is organized as follows: Section II discusses about the system and the flow of information during the communication over the network along with the security requirements for a strong protocol. Section III explains the proposed VASS-AKA protocol in detail. The security analysis of the proposed protocol is explained in section IV. Section V focuses on the performance evaluation of the proposed protocol. Section VI presents the conclusion of the paper.
A. System Model

We introduce a scenario for value added services, where multiple mobile stations (MS) send authentication requests to the authentication server (AS). It’s a challenge for AS to verify and authenticate maximum number of MS based on its capacity to handle authentication in an efficient way. This scenario is presented in the Figure 1 where multiple MS send their authentication requests to the AS at the same time or in a fixed duration time (which is very less). The AS handles all the authentication requests and sends the information of verified and authentic MS(s) to the server responsible for value added service. Afterward, the service provider server (SPS) provides the service to all valid MS(s). These authentication requests may be single or multiple, although there is very rare chance of single authentication request at a time to the AS.

If server handles one request at a time then it requires a queue to manage all the incoming requests. But management of a queue is becoming another task and increases overhead, execution time and cost of authentication. In fact, the AS processing and the approach used, must be very efficient to handle all the requests in very less time. One way for better handling multiple authentication requests at a time is to perform a batch authentication process for all the incoming requests at one time or in fixed time duration. But there may be one or more invalid request generated by an adversary. In such case first task is to identify the invalid request(s) and remove them from the batch, then perform re-batch authentication process. But here we have to pay additionally cost for every re-batch authentication process. Various notations used in the paper are illustrated in Table 1 along with their sizes.

B. Security Requirements

The following are the security requirements that must fulfill in order to develop a secure protocol for VAS.

1) Mutual Authentication

The developed protocol must provide mutual authentication, i.e. the MS must authenticate the valid AS to which it is requesting and the AS must verify the MS whether the MS is a part of this system and is a valid user. This mutual authentication process prevents system from the eavesdropping and impersonation attack.

2) Session Key Establishment

Secret session key is always a concern when we are dealing with a symmetric key based system. The protocol should be able to handle key generation, its transmission over the network and its usage.

3) Privacy Preservation

The original identity of each MS must be hidden during the transmission over the network. Such privacy preservation helps to prevent the system from Man-in-the-Middle and Redirection attacks.

III. PROPOSED PROTOCOL: VAS-AKA

This section proposed an efficient protocol named VAS-AKA for value added services as shown in the Figure 2.

A. System Assumptions

Here, we make some assumptions for this system which are following: (1) It is assumed that all the mobile stations (MS) have their unique identity as IMSI (International Mobile Subscriber Identity), (2) We consider the AuC (Authentication Center) as a part of the authentication server AS, (3) A key SK is stored at the AuC as well as in the SIM of MS at the time of manufacturing, (4) We assumed that the communication medium between the MS and the AS is secure enough for transmitting the information. If it is not, then it is required to provide integrity additionally. Both the MS and the AS can calculate the MAC (Message Authentication Code) of the information being sent at their end and can send it to each other along with the original message.

B. Batch Authentication and Key Agreement

In this subsection, we present an efficient authentication and key agreement protocol named VAS-AKA which provides a batch based mutual authentication between AS and the MS. Let ‘m’ be the total number of authentication requests generated by various mobile stations MS at the same time or in a fixed interval to the authentication server (AS). Initially, each MS choose a random number \( k_i \) \((1 < k_i < m)\), generate current timestamp \( T_i \) and \( DK_i \) where \( DK_i = f_i(SK, T_i) \). In fact, this \( DK_i \) is generated at both MS as well as the AS (later in this protocol) with a shared secret key \( SK \) which is stored at the AS and in the SIM card at the time of manufacturing. We consider that the authentication center (AuC) is a part of this authentication server (AS). Then every MS computes \( X_i = k_i \cdot IMSI_i \) and signature as \( S_i = (k_i + DK_i, O) \mod m \) where \((.)\) is bitwise X-OR operation. Each MS computes \( TID_i = f_i(IMSI_i, || T_i) \) to prevent the transmission of
1. Choose $k_i$, Generate $T_i$, and $DK_i$

2. Compute $X_i = k_i.IMSI_i$

3. $S_i = (k_i + DK_i, G) \mod m$

4. Compute $IMSI_i = f_2(TID_i \| T_i)$ and $DK_i$

5. Compute $P = \sum_{i=1}^{m} DK_i.IMSI_i$

6. Compute $R = \sum_{i=1}^{m} S_i.IMSI_i - G.P$

7. Compute $P_i' = DK_i.IMSI_i$

8. Check $P_i' = P_i$, if it holds then

AS is verified by all MS

The proposed VAS-AKA protocol successfully verifies the identities of the MSs over the network which obviates the ID theft, eavesdrop and Man-in-the-Middle attacks. Now, each MS sends an authentication request as $(T_i, X_i, S_i, TID_i)$ to the AS. On receiving the authentication requests, the AS computes $IMSI_i = f_2(TID_i \| T_i)$ and corresponding $DK_i$ with the help of $SK_i$. The function $f_2$ is just like any reversible symmetric encryption function where the plain text and shared key which is timestamp here, generate cipher text and the cipher text and same key (timestamp) are able to produce the original plain text. In the same way this function works to compute $IMSI_i$ and $TID_i$ with $T_i$. Next, the AS computes $P = \sum_{i=1}^{m} DK_i.IMSI_i$ and $R = \sum_{i=1}^{m} S_i.IMSI_i - G.P$, where 't' is the total number of invalid $MS_i$. Then the AS compares $\sum_{i=1}^{m} X_i = R$, if it holds then all the $MS_i$ are authenticated by the AS otherwise the AS initiates the re-batch authentication process. Finally AS sends all $P_i$ to the respective $MS_i$. All the $MS_i$ computes the $P_i'$ as $P_i' = DK_i.IMSI_i$ and compare it with the received $P_i$. If both are equal then the AS is verified by all the $MS_i$ otherwise the particular $MS_i$ terminates the connection and resend the authentication request to the AS.

C. Discussion

This subsection discusses about the reliability of the proposed protocol along with an algorithm to detect invalid $MS_i$ requests.

1) Reliability Analysis

In VAS-AKA protocol, if all $MS_i$ are successfully verified then this protocol achieves its maximum reliability with respect to its performance. In such case, this batch authentication scheme provides maximum successful authentications between
the AS and $MS_i$ at a time and generates minimum verification delay. Let $N_{AS}$ be the maximum number of authentication requests generated at a time. Out of these requests, some may be invalid authentication requests, assume them as $N_{IN}$. Since, $N_{AS}$ may be a very large number based on the type of value added service, the AS may not authenticate all the requests at one time due to its capacity. We assume that $N_{AS}$ is the maximum capacity of the AS to authenticate the requests at a time. For the statistical analysis, we assume that $N_{AS} = 1000$, $N_{IN} = 900$, and $N_{IN} = 1\%$ of the $N_{AS}$, i.e. 10. Let $prob\{t\}$ be the probability that exactly $t$ invalid authentication requests are sent to the AS. Then the probability of the Hypergeometric distribution is as follows:

$$prob\{t\} = \binom{N_{MS} - N_{IN}}{N_{AS} - t} \frac{N_{IN}}{t}$$

where $t = 1, 2\ldots 10$

This indicates that $(N_{MS} - t)$ valid requests are sent from the $(N_{MS} - N_{IN})$. One or more invalid request(s) in the batch leads to the batch verification failure and in such cases Re-batch verification is required.

2) Invalid Request Detection Algorithm

This subsection proposes an algorithm to detect the request from invalid $MS_i$ from a batch of authentication requests. The propose algorithm based on divide and conquer approach, is as follows: 

Input: The AS received a batch (AR) of $m$ authentication requests $\{R_1, R_2, R_3, \ldots, R_m\}$; Output: return the invalid request(s) otherwise return true.

Invalid_Req_Algorithm (AR):

1. if $(\text{Verify}(AR))$ then return True
2. else if $(\text{Size}(AR) == 1)$ then return IMSI_i $\in AR$ as invalid request.
3. else set $AR_1 = \{R_1, R_2, R_3, \ldots, R_{[m/2]}\}$
4. set $AR_2 = \{R_{[m/2]+1}, R_{[m/2]+2}, R_{[m/2]+3}, \ldots, R_m\}$
5. Invalid_Req_Algorithm ($AR_1$)
6. Invalid_Req_Algorithm ($AR_2$)

A batch of authentication requests can be divided at most $\lceil \log_2 m \rceil$ times. At the end, this algorithm has a set of total number of invalid requests from $MS_i$, and these invalid requests must be removed from the batch for the Re-batch authentication.

IV. SECURITY ANALYSIS

This section provides the security analysis of the proposed VAS-AKA protocol in terms of mutual authentication, session key establishment and privacy preservation.

A. Mutual Authentication

This VAS-AKA protocol provides mutual authentication between the AS and the $MS_i$. The AS authenticates the $MS_i$ by checking the $\sum_{i=1}^{M} X_i \equiv R$ and each $MS_i$ authenticates the AS by comparing the $P_i' \equiv P_i$.

B. Session Key Establishment

$DK_i$ key is used as a session key for each authentication between AS and the $MS_i$, which is generated from a shared secret key $SK_i$, stored at the AuC (part of AS) and in the SIM card at the time of manufacturing. This $DK_i$ key is generated as $DK_i = f_{1, MS_i}(T_i)$ where $T_i$ is the current timestamp when this key is being generated. The same key is used for a session within the expiry time.

C. Privacy Preservation

The privacy of each $MS_i$ is well protected during the authentication process over the network. $TID_i$ is computed from the original IMSI_i as $TID_i = f_2(\text{IMSI}_i \ || \ T_i)$ where function $f_2$ is reversible in nature.

V. PERFORMANCE EVALUATION

These sections analyzes the performance of the proposed VAS-AKA protocol in terms of mutual authentication, session key establishment, verification delay and re-batch verification delay.

A. Communication Overhead

The communication overhead can be defined as the total number of bits transmitted during the authentication process over the network. Here, we compare the communication overhead generated from various other protocols like IBV [1], ECDSA-AKA [2], BLA [4], and ABAKA [5] along with the VAS-AKA. We did not find any protocol directly related to our work in the literature. The above mentioned protocols provide authentication of value added services in vehicular ad hoc networks. However, we can compare communication and computation overhead generated by our protocol with these protocols because these protocols are based on mutual authentication and the flow of information is same in all the protocols. But the verification delay is different for vehicular ad hoc network protocols and cellular mobile network protocol such as VAS-AKA, because vehicular ad hoc network protocols have additional devices and road side equipments to communicate secure information in the network. Now, we
discuss the transmission overhead during the single as well as multiple authentication request process.

**Single Authentication:** When \( n=1 \) in a batch authentication process, then it is a single authentication request process where ‘\( n \)’ is the maximum number of authentication requests at a time. Table 2 explains the total number of bytes needs to be transmitted in different protocols from device to server and server to the device. In the proposed VAS-AKA protocol, the device is a mobile station and server is the authentication server. It can be clearly observed that out of IBV, ABAKA, BLA, ECDSA-AKA, VAS-AKA, the proposed VAS-AKA protocol generates minimum transmission overhead, i.e. 56 and 16 bytes from MS to AS and AS to MS respectively.

**Multiple Authentications:** Multiple authentication requests ‘\( n \)’ can be handled by the single batch verification process at a time. But, it definitely increases the computation and overhead. Table 3 represents the number of transmitted bytes over the network to complete the \( n \)-authentication requests generated by different mobile stations. The communication overhead generated by \( n \)-authentication request process is equal to the \( n \)-communication overhead generated by a single authentication request process. Both the authentication processes can be analyzed through Figure 3 which represents two separate graphs, one for the device to the server and other for a server to the device. We can observe that in both the graphs VAS-AKA protocol generates minimum transmission overhead. (In server to device graphs, IBV and BLS overheads assume to zero as they do not apply this process during the authentication.) Figure 4 and Figure 5 each shows a graph for one way multiple authentications between MS authentication requests and size of transmitted bytes from device to server and from server to device respectively. Both Figures clearly indicate that VAS-AKA produces a minimum number of transmitted bytes during the ‘\( n \)’ authentication request process where \( n = 50, 100, 200, 500 \) and 1000. In Figure 5, IBV and BLS protocols do not send any information (byte) from the server to the device, thus have considered zero values. In single authentication as well as multiple authentication process, the proposed VAS-AKA protocol is compared with the other discussed protocols.

![Figure 3. Single Authentication](image3.png)

![Figure 4. One way Multiple Authentication: Device to the Server](image4.png)

![Figure 5. One way Multiple Authentication: Server to the Device](image5.png)

In one way multiple/single authentication request(s) process from device to the server, the VAS-AKA protocol is able to reduce 11.12%, 33.34%, 61.65%, and 66.47% transmission bandwidth as compared to the IBV, ABAKA, BLS and ECDSA-AKA protocols respectively. Similarly, in the transmission of bytes from server to the device for one way multiple/single authentication request(s) process, the VAS-AKA lower the bandwidth as 80% and 90.5% as compared to the ABAKA and ECDSA-AKA protocols.

**B. Computation Overhead**

This subsection analyzes the computation overhead generated during the authentication process in the VAS-AKA protocol. Here, we consider both types of authentications, i.e. single authentication request process and the multiple authentication request process (in batch).
Single Authentication \((i = 1)\): The following computations are done at the MS and the AS during the single authentication.

**Computation at the MS:** (1) \(f_2(\text{IMSI} || T_f)\), (2) \(f_{1_{sk}}(T_s)\), (3) \(k_i \cdot \text{IMSI}_i\), (4) \(Y = G \cdot Dk_i\), (5) \(k_i + Y\), (6) \(Dk_i \cdot \text{IMSI}_i\); **Computation at the AS:** (1) \(f_{1_{sk}}(T_f)\), (2) \(f_2(\text{IMSI} || T_f)\), (3) \(Dk_i \cdot \text{IMSI}_i\), (4) \(S_i = S_i \cdot \text{IMSI}_i\), (5) \(G' = G \cdot P\), (6) \(S' = G'\)

**Multiple/Batch Authentication** \((i = 2, 3... m)\): The computation performed during the multiple authentications is as below:

**Computation at the MS:** (1) \(m[f_2(\text{IMSI} || T_f)]\), (2) \(m[f_1_{sk}(T_s)]\), (3) \(m[k_i \cdot \text{IMSI}_i]\), (4) \(m[Y = G \cdot Dk_i]\), (5) \(m[k_i + Y]\), (6) \(m[Dk_i \cdot \text{IMSI}_i]\); **Computation at the AS:** (1) \(m[f_2(\text{IMSI} || T_f)]\), (2) \(m[f_{1_{sk}}(T_f)]\), (3) \(m[Dk_i \cdot \text{IMSI}_i]\)

For AS: \(m \cdot T_{mul}\); Verification Delay = \(m \cdot T_{add}\) nanoseconds

**D. Re-batch Verification Delay**

If the batch authentication process is not successful then it requires Re-batch authentication without including the detected invalid MS\(_i\). The total operations required in re-batch authentication are:

\[(m-t)[P = \sum_{i=1}^{m} P_i], (m-t)[S' = \sum_{i=1}^{m} S'_i], G' = G \cdot P, (m-t)[X_i], and S' = G' = 3(m-t) \cdot T_{add} + T_{mul} + T_{sub}\]

where \(t\) is the no of invalid MS\(_i\), which have been removed in Re-batch authentication.

Verification Delay Time = 2799*(m-t) + 31255 nanoseconds

**VI. Conclusion**

The proposed protocol provides mutual authentication between each MS and the AS. This protocol efficiently verifies multiple authentication requests at a time or in a fixed (very less) time duration and keeps the original IMSI private during the authentication process. The authors claim that this is the first batch oriented AKA protocol which provides value added services to the mobile users. In one way multiple/single authentication request(s) process from device to the server, the VAS-AKA protocol reduces 11.12%, 33.34%, 61.65%, and 66.47% transmission bandwidth as compared to the IBV, ABAKA, BLS and ECDSA-AKA protocols respectively. From server to the device one way multiple/single authentication request(s) process, the VAS-AKA lowers the communication bandwidth by 80% and 90.5% as compared to the ABAKA and ECDSA-AKA protocols.

**Acknowledgment**

This work is supported by Tata Consultancy Services Limited (TCS), India.

**References**


